

App de sistema de consulta, creación y préstamos de libros.

**Dependencias utilizadas:**

- **body-parser** “^1.19.0”: Middleware de análisis del cuerpo de node.js
- **express** “^4.17.1”: Es un framework web muy popular de node.js, y es la librería subyacente para un gran número de otros frameworks web de node.js populares.
- **mongoose** “^3.6.3”: El controlador oficial de MongoDB para node.js. Proporciona un api de alto nivel.
- **pug** “^3.0.0”: Pug (anteriormente llamado jade) es un motor de plantillas de alto rendimiento fuertemente influenciado por Haml e implementado con JavaScript para node.js y navegadores.  
**NOTA:** Se usó pug en el código HTML debido a su facilidad y redundancia de código mínima que se pudo llegar a generar, además de sus prácticas funciones adicionales para un mejor manejo por parte del programador.
- **nodemon** “^2.0.6”: Es una herramienta que ayuda en el desarrollo de apps basadas en node.js al reiniciar automáticamente la app de nodo cuando se detecten cambios de archivo en el directorio.

**Puerto:**

Esta app correrá en el puerto 3000.

**Archivo index.js:**

Este contiene conexiones con express, body-parser, pug y con la base de datos de mongoose.

Estructura del archivo:

Para la colección libros:

**(/l) (GET)**

Esta devolverá una vista de los documentos almacenados en la colección libros, si esta se encuentra vacía entonces devolverá un mensaje especificando esto.

**(/agregar\_lib)(GET)**

Esta devuelve la interfaz para registrar libros.

#### **(/agregar\_lib)(POST)**

Esta insertará los datos guardados al documento en la colección libros.

#### **(/mostrar/:id)(GET)**

Esta devuelve los datos del documento previamente guardados en la colección libros y permite visualizarlos.

#### **(/actualizar/:id)(GET)**

Esta devuelve la interfaz para actualizar libros, además nos traerá los datos del documento previamente guardados en la colección libros para poder modificarlos si se requiere.

#### **(/actualizar\_lib)(PUT)**

Esta guardará los datos que se le modificaron previamente a la colección libros en su respectivo documento.

#### **(/eliminar\_lib)(DELETE)**

Esta eliminará los datos del documento seleccionado, sin afectar a los demás documentos de la colección libros.

Para la colección prestamo:

#### **(/l) (GET)**

Esta devolverá una vista de los documentos almacenados en la colección prestamo, si esta se encuentra vacía entonces devolverá un mensaje especificando esto.

#### **(/agregar\_pres)(GET)**

Esta devuelve la interfaz para registrar prestamos.

#### **(/agregar\_pres)(POST)**

Esta insertará los datos guardados al documento en la colección prestamo.

#### **(/mostrar\_/:id)(GET)**

Esta devuelve los datos del documento previamente guardados en la colección prestamo y permite visualizarlos.

### **(/actualizar\_/:id)(GET)**

Esta devuelve la interfaz para actualizar libros, además nos traerá los datos del documento previamente guardado en la colección prestamo para poder modificarlos si se requiere.

### **(/actualizar\_pres)(PUT)**

Esta guardará los datos del documento que se le modificaron previamente a la colección en su respectivo documento.

### **(/eliminar\_pres)(DELETE)**

Esta eliminará los datos del documento seleccionado, sin afectar a los demás datos de la colección prestamo.

## **CARPETA BIBLIOTECA**

Estructura de la carpeta:

Biblioteca

node\_modules

views

css

style.css

libros

actualizar\_lib.pug

agregar\_lib.pug

libros.pug

mostrar\_lib.pug

prestamos

actualizar\_pres.pug

agregar\_pres.pug

prestamos.pug

mostrar\_pres.pug

index.pug

index.js

package-lock.json

package.json